AD-A285 966

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

94-34404

A PROTOTYPE DECISION AID FOR
ESTIMATING SALVO DAMAGE EFFECTS
BASED ON A CELLULAR MODEL

by

Jeffrey D. Varady

September, 1994

Thesis Advisor: James Esary

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE **A PROTOTYPE DECISION AID FOR ESTIMATING SALVO DAMAGE EFFECTS BASED ON A CELLULAR MODEL** (U)

5. FUNDING NUMBERS

6. AUTHOR(S) Varady, Jeffrey D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(maximum 200 words)*

This study addresses the development of a Tactical Decision Aid to assess expected damage to a target from a salvo of warheads. It is based on a recently developed Cellular Target Concept. A secondary purpose for the development of the TDA was its potential use as an investigatory tool. Previous work with cellular targets has been confined to models whose characteristics lead to simple mathematical solutions. Many target models do not lead to simple solutions. There has been some interest in observing if these models resemble the simple models asymptotically. The TDA has been designed to allow for a better understanding of how damage aggregates in these more complex models, especially when compared to the proportional damage aggregation observed in many of the simpler models. The comparisons yielded some surprising results. None of the models designed to test asymptotic proportionality appeared to show this property in the long run. Some theories are discussed in the study. The theoretical tools used to test the asymptotic behavior of the models are discussed in an appendix.

14. SUBJECT TERMS Cellular Target Modeling, Tactical Decision Aid

15. NUMBER OF PAGES 78

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

A Prototype Decision Aid For Estimating Salvo Damage Effects Based on a Cellular Model

by

Jeffrey D. Varady
Lieutenant, United States Navy
B.S., University of Wisconsin, 1986

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1994

Author: Jeffrey D. Varady
Jeffrey D. Varady

Approved by: J. Esary
James Esary, Thesis Advisor

Lyn R. Whitaker
Lyn R. Whitaker, Second Reader

R R Read
for Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

This study addresses the development of a Tactical Decision Aid to assess expected damage to a target from a salvo of warheads. It is based on a recently developed Cellular Target Concept. A secondary purpose for the development of the TDA was its potential use as an investigatory tool. Previous work with cellular targets has been confined to models whose characteristics lead to simple mathematical solutions. Many target models do not lead to simple solutions. There has been some interest in observing if these models resemble the simple models asymptotically. The TDA has been designed to allow for a better understanding of how damage aggregates in these more complex models, especially when compared to the proportional damage aggregation observed in many of the simpler models. The comparisons yielded some surprising results. None of the models designed to test asymptotic proportionality appeared to show this property in the long run. Some theories are discussed in the study. The theoretical tools used to test the asymptotic behavior of the models are discussed in an appendix.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While effort has been made, within the time available, to ensure that the programs are free of computational logic and errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This study addresses the development of a prototype Tactical Decision Aid to assess expected damage to a target from a salvo of warheads. The TDA is based on a recently developed cellular target concept. A computer based (TDA) can assist a strike planner to rapidly estimate expected damage to a target. Affordable PC based computers, coupled with the improved and modestly priced spreadsheet programs now available, appear to offer an ideal venue in which to develop such a TDA.

The study demonstrates the potential of using spreadsheets to develop decision aids of this type. It is often argued that the operational commander can not utilize operational research techniques because of the need to make rapid decisions during changing conditions. As personal computers continue to be made available, it will be vital that programs of this type are developed. They can bridge the perceived gap between analysts and operators. This TDA has the potential to be continually modified to enhance its value to the user.

There are a couple of aspects of the decision aid that improve the ability of the user to determine damage to a

target. First, the TDA allows the user to visualize the target as the model is built. Another strength is the ability of the TDA to either determine expected damage from a known salvo size or determine the number of warheads required to achieve a desired level of damage. This allows the user different ways to plan an attack.

The TDA allows the user to divide a target into cells. It will accommodate up to 100 cells in a 10X10 grid. The TDA incorporates three internal submodels; a Hit Distribution Model, a Warhead Pattern Dispersion Model and a Cellular Hardness Model.

There are two potential shortcomings in the prototype. Both of these are results of the spreadsheet base for TDA. The first involves the speed of the TDA. The second is round off error. Spreadsheets have not often been utilized in the fashion of this study.

This does not limit the TDA when performing what it is advertised to do. The calculation of total expected aggregate damage can be as accurate as one performed on a scientific calculator.

A secondary purpose for developing this decision aid was its potential use as an investigatory tool. The TDA has been designed to calculate two indices that may allow for a better understanding of how damage aggregates in more

complex models, especially when compared to the proportional damage aggregation observed in many of the simpler models.

Many targeting scenarios do not have a structure that leads to proportional damage aggregation. It seems intuitive that some of these scenarios should begin to show "proportionality" as the number of warheads impacting the target increases beyond some threshold level. To study this the TDA has been designed to calculate two parameter sequences, indices of proportionality for the target for given numbers of hits and damage increment rates.

The TDA was executed using nine different scenarios based on a five cell target. This was done for three reasons. Two were to confirm the accuracy of the TDA for models with known analytical solutions. This was accomplished. The other was to explore the behavior of the two parameter sequences in some more complex models.

It was theorized that the more complex targeting models would show proportional damage aggregation in the asymptotic sense. It appears that none of the complex scenarios show proportional damage aggregation, however the calculations performed to find the research parameters are perhaps better accomplished on any of the large mathematical programming packages available.

I. INTRODUCTION

There has long been military interest in modeling the effects of weapons on targets. A related subject is the effects of weapons salvos on a target. An attacker could address the question of expected damage to a target in two different ways. First, one may wish to consider how much damage would be inflicted if a fixed number of warheads impact the target. A second way is to consider how many warheads are required to destroy a fixed percentage of the target. A computer based Tactical Decision Aid (TDA) would be highly valuable in assisting the strike planner to rapidly estimate the expected damage to a target. A prototype Tactical Decision Aid has been developed based on one approach to this capability.

The recent explosion of affordable PC based computers has resulted in most every military command having several of these computers. These powerful machines, coupled with the improved and modestly priced spreadsheet programs now available, appeared to offer an ideal venue in which to develop this TDA.

This study addresses the development of a Prototype Tactical Decision Aid to assess expected damage to a target

from a salvo of warheads. The TDA is based on the Cellular Target Concept recently developed by Esary(Esary, 1990, pp. 16-17).

The TDA was implemented using Microsoft Excel for Windows Version 5.0. Excel was chosen for its widespread use and affordability. Additionally, Excel is available for both DOS and Apple based computers.

A secondary purpose for developing this decision aid was its potential use as an investigatory tool. Previous work with cellular targets has only been done on models whose characteristics lead to simple mathematical solutions. The TDA was tested for accuracy using target models for which expected damage can be determined from closed form equations.

Many target models do not lead to simple solutions. There has been some interest in observing if these targets tend toward a simple solution asymptotically. The TDA has beer designed to calculate two indices that may allow for a better understanding of how damage aggregates in these more complex models, especially when compared to the proportional damage aggregation observed in many of the simpler models(Esary, 1990, pp. 15-16).

Chapter II will briefly discuss the cellular target concept. It will also describe the Tactical Decision Aid, its development and data made available to the user.

Chapter III will develop the theory of proportional damage aggregation and the models used to study this. Chapter IV will discuss the results of this study. Chapter V develops the conclusions of the analysis and some recommendation for possible follow on work.

II. THE TACTICAL DECISION AID

A. CELLULAR TARGET MODELS

The goal of this decision aid is to determine the effect of multiple weapons impacting a target. A major difficulty in calculating aggregate damage to a target is due to the often overlapping areas of damage caused by warheads as they impact the target. An approach to overcoming this obstacle is to view targets as being decomposed into cells. This cellular modeling approach divides the target into disjoint cells. A warhead that impacts the target can cause damage to one or more cells. Probable damage to individual cells can then be determined. Once this is accomplished, $E(D)$ - total expected aggregate damage to the target, is relatively simple to calculate.(Esary, 1990, pp. 15-25)

B. DESCRIPTION OF THE DECISION AID

The TDA allows the user to divide a target into cells. It will accommodate up to 100 cells in a 10X10 grid. A cell that is part of the target has these features:

- A user defined relative value.[1]
- A probability that the cell is hit by a warhead that reaches the target.
- The cell's hardness.

The target is modeled on a 100 cell area on a page of the spreadsheet. It allows the user a crude two dimensional view of the target that might, but need not, reflect the actual geometry of the target. As cells are added to the target, that cell changes color from red to yellow. The TDA will query the user upon entering the program whether the user desires to preset all the cells to the same parameters. Additionally, cells can be individually manipulated at any time prior to the execution of the calculations. This allows the user to build the desired model. For detailed instructions on the use of the TDA see Appendix A.

C. DECISION AID INTERNAL MODEL

The Tactical decision aid was designed using three internal submodels:

- A Hit Distribution Model
- A Warhead Pattern Dispersion Model
- A Cellular Hardness Model

---

[1] The relative cell value is entered by the user and is completely arbitrary. It is simply a value of the cell relative to all other cells in the target.

The Hit Distribution Model determines how many warheads fired in a given salvo actually reach the target. For this decision aid this submodel was assumed to have a binomial distribution. The user enters the number of warheads in the salvo, n, and the probability a warhead hits the target. The number of warheads that hit the target is a random number N with possible values $0,1,2, \ldots, n$.

The Warhead Pattern Dispersion Model is utilized to determine which cells in the target area an individual warhead impacts. For this decision aid each warhead was assumed to impact on one cell chosen according to a multinomial distribution. The probability that an individual cell is hit, r_i, is entered by the user. The sum of the r_i's must equal 1.0.

The Cellular Hardness Model determines the ability of an individual cell to withstand the impact of warheads upon it. The parameters for this model are entered by the user as cumulative probabilities that the cell will become damaged on the j^{th} hit. The program allows an individual cell to withstand up to 10 impacts before it is damaged with probability 1.0.

Once the user is satisfied with the target model, the decision aid is executed. The following calculations are displayed:

- n - the number of warheads fired at the target

- $E(D)$ - Total Expected aggregate damage.

The following calculations are also available to the user.

These are displayed for each k = 1, ..., n:

- $\delta_i(k)$ - the probability the i^{th} cell is damaged given k hits on the target.
- $D(k)$ - the proportion of damage to the target given k hits on the target.

- d_k - an index for the proportion of damage per hit. This is described in Chapter III.

- a_k - A damage increment rate for the k^{th} hit. This is also described in Chapter III.

The last two listed, d_k and a_k, are used to study some aspects of cumulative damage aggregation.

III. USING THE DECISION AID

A. CUMULATIVE DAMAGE AGGREGATION

Let $D(k)$ be the proportion of damage to a pristine target from exactly k hits. If the proportion damaged from a single hit is d and each additional hit damages the same proportion d of that part of the target not previously damaged then,

$$D(k) = 1 - (1 - d)^k$$

This has been discussed as a proportional effects mechanism for aggregating the cumulative effects of hits.(Esary, 1990 p. 15) Several targeting scenarios have been shown to result in proportional damage aggregation.

Many targeting scenarios do not have a structure that leads to proportional damage aggregation. A scenario which requires more than one hit before there is a positive probability of damaging the target is an example of a scenario that is not proportional. It seems intuitive that some of these scenarios should begin to show "proportionality" as the number of warheads impacting the target increases beyond some threshold level.

To study this the TDA has been designed to calculate two parameters, d_k and a_k. As previously mentioned, d_k is an index of proportionality to the target given k hits. It is calculated using the same formula satisfied by d when there is proportionality,

$$D(k) = 1 - (1 - d_k)^k .$$

This is solved for d_k yielding,

$$d_k = 1 - (1 - D(k))^{1/k} .$$

The parameter a_k is a damage increment rate. It calculates the proportion of the previously undamaged target that is damaged by the k^{th} hit. It is defined by an equation also satisfied by d when there is proportionality,

$$D(k) = D(k - 1) + a_k[1 - D(k - 1] .$$

Solving this equation for a_k yields,

$$a_k = \frac{D(k) - D(k - 1)}{1 - D(k - 1)} .$$

A detailed discussion of the relationships between d_k and a_k can be found in Appendix B.

If a targeting scenario does lead to asymptotic proportionality, it appears intuitive that both d_k and a_k should approach constants as the previously mentioned threshold is exceeded.

B. MODEL USED

The TDA was executed 27 times using nine different scenarios based on a five cell target. This was done for three reasons. The first reason was to confirm that the TDA was accurately calculating the sequences of d_k and a_k for scenarios that were known to have proportional damage aggregation. Since these targets have simple mathematical solutions for $D(k)$, it was possible to confirm the accuracy of the TDA in calculating $D(k)$ for these models. Three scenarios were designed for this purpose.

The second purpose for the runs was to explore the behavior of d_k and a_k for the more complex models discussed previously. Six scenarios were designed for this purpose. Each of these scenarios was run one time with a salvo size equal to 100 and a probability the target was hit equal to 0.6. The large salvo size ensured complete sequences for d_k and a_k.

The final reason for the runs was ensure the TDA would provide the user with the same solution regardless of which calculation option was chosen. The two options available to the user are:

- Calculate $E(D)$ to the target for a given salvo size and probability a warhead reaches the target.
- Determine the salvo size required to achieve a predetermined $E(D)$ for a given probability a warhead reaches the target.

The TDA must generate complimentary solutions for the options available. For example, if $E(D)$ is calculated for a target based on 10 hits, the TDA must be able to use that calculated $E(D)$ as the input for determining salvo size and tell the user that 10 warheads are required. The previous statement is true only if the probability a warhead reaches the target is the same when running both options. Testing this was accomplished by running the nine scenarios two more times each as described above. They were first run with a salvo size of 10 and probability of hit equal to 0.6. Those outputs were then the inputs to run the scenarios a second time.

C. SCENARIO DESCRIPTION

The following is a general description of the nine targeting scenarios designed to be analyzed by the TDA. A detailed description of each of the scenarios can be found in Appendix C.

Scenario 1 is a CE11UI targeting scenario.[2] Each cell is equally valued and equally likely to be hit. The hardness model is the same for each cell. Each cell is damaged by one hit.

---

[2] This method for describing cellular targeting models was developed in *A Basic Lemma on Expected Damage Aggregation for Cellular Targets, and Some of its Applications* (Esary, 1991, pp. 18-21).

Scenario 2 is a CZ11UI scenario. Four cells are equally valued, one has a relative value of zero. The remainder of the model is identical to Scenario 1.

Scenario 3 is a CA11UI scenario. The relative values of the five cells are not identical. The remainder of the model is identical to Scenario 1.

Scenario 4 is a CE12UI scenario. It is identical to Scenario 1 except it takes at two hits to damage any cell.

Scenario 5 is a CA12UI scenario. It differs from Scenario 4 in the relative values, w_i, of the cells. One of cells is six times more valuable than the other four equally valued cells.

Scenarios 6 and 7 are very similar. The relative values of the cells are the same as in Scenario 5. The probability a given cell is hit by a warhead is the same for each cell. In Scenario 6 each cell is damaged by a single impact, while it takes two impacts to damage a cell in Scenario 7.

Scenarios 8 and 9 are designed as CE1PrUI scenarios. Pr indicates each cell's hardness is modeled as a probability function. All the previous scenarios require a discrete number of impacts to inflict damage to a cell. In Scenario 8 all cells are equally weighted and may be damaged by one through five hits. Scenario 9 is identical to Scenario 8 except it takes at least two hits on any cell before it can be damaged.

IV. RESULTS

The 27 runs of the TDA were completed as described in the previous chapter. The results are divided into three sections. The first section discusses the known solutions. The second section discusses the TDA's ability to generate the complimentary solutions discussed above. The third section will discuss the six runs that were completed on the more complex models and the behavior of the sequences of d_k and a_k for these scenarios.

A. COMPARISONS TO KNOWN SOLUTIONS

Scenarios 1, 2 and 3 were constructed to exhibit proportional damage aggregation. The TDA showed that d_k for all k was equal to d. The same was true for a_k. The TDA maintained this accuracy throughout the range of $D(k)$. When $D(k)$ was equal to 1, the sequences were complete and the rest of the calculations were ignored.

Previous work has shown that the first three scenarios have known values for d and a (Esary, 1991 pp. 16-19). Table 1 has the values calculated using the known formulae and the values calculated by the TDA.

TABLE 1. COMPARING d AND a TO KNOWN SOLUTIONS

Scenario	Known d	Calculated d	Known a	Calculated a
1	$d=1/m=0.2$	0.2	0.2	0.2
2	0.2	0.2	0.2	0.2
3	0.2	0.2	0.2	0.2

Using the equation for calculating $D(k)$ for a target that exhibits proportional damage aggregation, is was a simple matter to confirm that the TDA was accurately calculating $D(k)$ for the scenarios above. Total expected aggregate damage, $E(D)$, can be calculated for these models by the equation,

$$E(D) = \sum_{k=0}^{n} D(k)P[N = k].$$

This was done for the first three scenarios and yielded the results found in Table 2.

TABLE 2. COMPARING $E(D)$

Scenario	Calculated $E(D)$	Known $E(D)$
1	.999998	1.0
2	.999998	1.0
3	.999998	1.0

The calculations were for a salvo size of 100 against a five cell target. The main purpose of these first runs was to confirm that the TDA was accurately generating complete sequences of d_k and a_k. The calculations of $D(k)$ and $E(D)$ are compared again in Section B when these same scenarios are run again with a smaller salvo size.

B. TESTING COMPLIMENTARY SOLUTIONS

Each scenario was executed two times. The first run was to determine the expected damage to the target. The second run used the $E(D)$ given as the result of the first run as input to determine the number of warheads required to accumulate that level of damage. These 18 runs complimented each other exactly. The results can be seen in Appendix D.

As previously discussed, the first three scenarios display proportional damage aggregation and have known simple mathematical solutions available for $D(k)$ and $E(D)$. The results were calculated for the scenarios using the parameters discussed. The results are in Table 3.

TABLE 3. COMPARING $D(k)$ AND $E(D)$, SALVO SIZE = 10.

Scenario	TDA Calculated $E(D)$	Known $E(D)$
1	.721499	.721499
2	.721499	.721499
3	.721499	.721499

C. ANALYZING THE MORE COMPLEX MODELS

It was theorized that the more complex targeting models would show proportional damage aggregation in the asymptotic sense. Scenarios 4 through 9 were designed to test this hypothesis. The results of the 100 warhead salvo size runs of these scenarios can be found in Appendix E.

It appears that none of the complex scenarios show proportional damage aggregation. Four of the six scenarios are targets where r_i is equal for all the cells. These scenarios had sequences of d_k and a_k that were monotonically increasing. This begins to make sense. As warheads continue to impact the target, they will continue to inflict damage. The sequences begin to level off as $D(k)$ increases toward 1.0. So damage is continuing to aggregate but at a decreasing rate. It just does not appear to reach a steady state.

Scenario 8 is a particularly puzzling scenario. This was designed to be a generalization of scenarios that display proportional damage aggregation. The only difference in this scenario is the hardness model for the cells. In Scenario 8 the cell hardness is probabilistic instead of deterministic. There is a positive probability that a cell can be damaged by a single hit. It seemed that if any of the more complex scenarios would show asymptotic proportionality, it would be this one. Even this scenario did not behave as expected.

Two of the scenarios also displayed unexpected behavior. Scenarios 6 and 7 have one cell that is more likely to be hit than the others. The sequences of d_k and a_k for these scenarios increased to a maximum then began to decrease. This seems logical. As warheads continue to impact the

target they will tend to hit that high probability cell. Eventually that cell is damaged. Warheads will continue to impact that same cell rather than any of the other cells. Since that cell can no longer contribute to an increase in the total damage to the target, the rate at which damage increases to the target will diminish.

V. CONCLUSIONS

This TDA was designed as a prototype to explore the possibility of implementing cellular targeting methodology on a spreadsheet. It has shown some strengths and potential shortcomings. This chapter will discuss these and then include some areas for potential study.

A. STRENGTHS

There are a couple of aspects of the decision aid that improve the ability of the user to determine damage to a target. First, the TDA allows the user to visualize the target as the model is built. Many times just picturing the target is useful. One can be sure the model portrays the target as accurately as possible. The model can be fine tuned to ensure it is doing this.

Another strength is the ability of the TDA to either determine $E(D)$ from a known salvo size or determine the number of warheads required to achieve a desired level of damage. In trying to decide how many bombers to launch against a target, it is not necessary to manually input a sequence of salvo sizes to the decision aid. The TDA will do this automatically. Alternatively, if the decision

about how many bombers will be launched is known, then the option that will tell how much damage is expected to be inflicted can be exercised.

B. POTENTIAL SHORTCOMINGS

There are two potential shortcomings in the prototype. Both of these are results of the spreadsheet base for the TDA. The first involves the speed of the TDA. The spreadsheet is capable of performing very rapid calculations. The slowdown appears to occur when it is necessary to read or write a value to a sheet in the file. If values are retained internally as variables, the program appears much faster. This was done as much as practicable.

The second concern is round off error. Spreadsheets have not often been utilized in the fashion of this study. The calculations performed to find d_k and a_k are perhaps better accomplished on any of the large mathematical programming packages available.

This does not limit the TDA when performing what it is advertised to do. The calculation of $E(D)$ was as accurate as one performed on a scientific calculator. It is also not necessary for the user to have much more than two decimal place accuracy for this number.

C. POTENTIAL FOLLOW ON STUDY

A couple of areas of study came to mind during this study. They were considered outside its scope. They can be pursued using or modifying the TDA.

First, there has been very little discussion concerning the relative values of the cells in a target. The study applied values arbitrarily. It could be valuable to find real targets and do a study to determine more accurate methods of valuing cells. These improved methods could then be incorporated into the TDA. A library of generic target types could then be available to a decision maker. This could be valuable as a time saver.

Second, the macros in the TDA could be modified to relax the assumptions made in this study. The Hit Distribution Model and the Warhead Pattern Dispersion Model both have distributional assumptions that limit the types of scenarios that can be examined. These assumptions could be relaxed to allow many more scenario types. The macro language in Microsoft Excel is Visual Basic for Applications. It is a fairly easy language to learn. A copy of the macro code can be found in Appendix F.

VI. RECOMMENDATIONS

This study demonstrates the potential of using spreadsheets to develop decision aids of this type. It is often argued that the operational commander can not utilize operational research techniques because of the need to make rapidly decisions during changing conditions. As personal computers continue to be made available, it will be vital that programs of this type are developed. They can bridge the perceived gap between analysts and operators. This TDA has the potential to be continually modified to enhance its value to the user.

APPENDIX A. TDA USERS MANUAL

This users manual provides the user with step by step instruction on the use of the Tactical Decision Aid. It assumes the user has a working knowledge of Microsoft Excel for Windows version 5.0. The users manual for Excel should be referred to for any question concerning operations in Excel that are not covered in this manual.

A. GENERAL DESCRIPTION

The Tactical Decision Aid is designed to model a target of up to 100 cells. It gives the mission planner two options for planning a strike. First, Expected Damage can be determined for a given salvo size. Second, the number of warheads required to reach a given damage level can be determined.

The ***Target*** worksheet provides the user with a 10X10 view of the target area. Cells that are part of the target are colored yellow and cells that are not are colored red. Each cell has a button that is pressed to change its attributes.

Most of the interactions between the user and the TDA are completed using Dialog Boxes. The programmed logic

requires that all the relevant boxes in the dialog box must be filled in prior to execution of a dialog box. To execute a dialog box the OK button must be depressed.

B. DEFINITIONS

The following are several definitions that will be required to use the TDA.

- **On Cell** - A cell that is part of the target area. On Cells are assigned a relative value, a probability of hit and a cell hardness model.

- **Off Cell** - A cell that is not part of the target.

- **Relative Cell Value** (w_i) - The value of the i^{th} cell relative to all other cells in the target. For the TDA to operate Σw_i must equal 1.0.

- **Probability Cell Hit** (r_i) - The probability a warhead that reaches the target will impact that particular cell. For the TDA to operate Σr_i must equal 1.0.

- **Cell Hardness** - The ability of a cell to withstand the impact of warheads upon it. A Dialog Box allows the user to enter a cell hardness as a cumulative probability function P[cell i damaged| j hits]. The cell can be modeled to withstand up to 10 hits before it is damaged with probability 1.0.

- **Salvo Size** - The number of warheads launched at the target.

- **Probability Hit** - The probability that a round fired at the target reaches the target.

C. **GETTING STARTED**

1. From Windows, enter Excel.

2. Open workbook TDA.XLS

3. A Dialog Box will appear that will give the user the option of reinitializing the Target or continuing with previous work. Choose one option and press enter. See Figure 1.

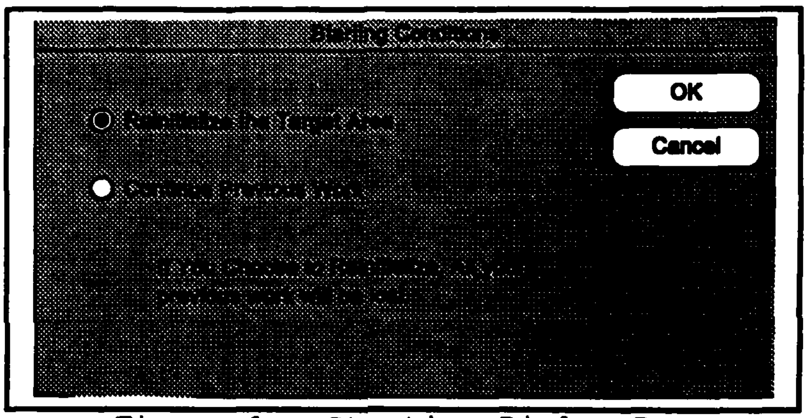

Figure 1. Starting Dialog Box

D. **CHOOSING REINITIALIZE OR CONTINUE**

If one chooses to Reinitialize, the user is queried whether all the cells are to be turned On or Off.

1. If Cells Off is chosen, a 10X10 target area is presented to the user with 0 active cells.

2. If Cells On is chosen, the 100 cells will be On, all cells will be given a relative value of 0.01 and the probability each cell is hit is set to 0.01. A dialog box will appear after the user enters the Cell On option. This

is for entering the cell's hardness. See Section F for details on entering a cell's hardness.

E. MAKING CHANGES TO INDIVIDUAL CELLS

Each cell in the Model has a button on top of it. This is pressed to make any of the following changes to a cell:

- Turn a cell On or Off
- Change the cell's relative value
- Change the cell's probability of hit
- Change the cell's hardness.

1. When the cell's button is depressed, a dialog box appears. It gives the user the option of turning the cell Off or On. In order to properly turn the cell on, the relative value and the probability of hit must be correctly filled in also. See Figure 2. A dialog box will appear to enter the cell's hardness when OK is pressed.

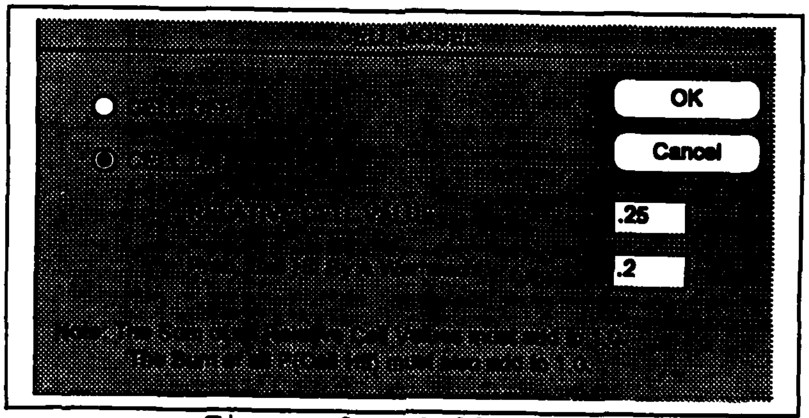

Figure 2. Cell Model

F. MODELING HARDNESS

A cell's hardness is modeled as a cumulative probability function. The ***Hardness Dialog*** has the user enter the probability cell *i* is damaged given *j* hits on that cell (P[cell *i* damaged| *j* hits]). Once the cumulative probability equals one, the user must ensure that all the subsequent entries are 1. For example, if cell i is destroyed after 6 hits, the boxes for seven through 10 hits must also be set to 1. See Figure 3.

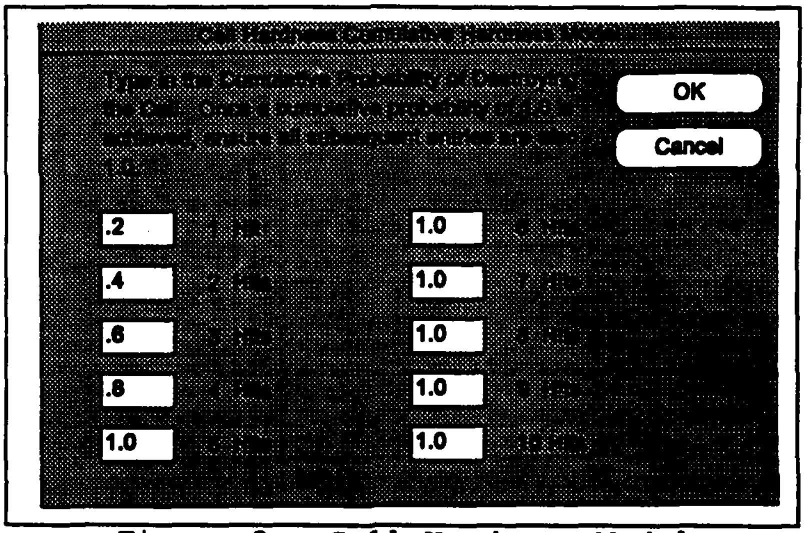

Figure 3. Cell Hardness Model

G. EXECUTING THE TDA

There are two ways to execute the TDA once the user is satisfied with the model. Option 1 is to determine $E(D)$. Option 2 is to run to a desired $E(D)$.

1. **Determine *E(D)*.** If this option is desired, depress the button marked ***Determine E(D)***. A dialog box appears. This queries the user for the number of warheads fired and the probability a warhead hits the target.

2. **Run to desired *E(D)*.** To execute this option, depress the button marked ***Run to Desired E(D)***. A dialog box appears. The user must enter the desired $E(D)$ and the probability that a warhead hits the target.

H. OUTPUT

The TDA will display the ***Results*** worksheet when it has completed the calculations. The results presented are:

- $E(D)$
- The number of warheads fired.

The expected proportion of damage to the target given k hits on the target ($D(k)$) is calculated by the program and the results are placed on the ***D(K)*** worksheet. Additionally, d_k and a_k are presented on this worksheet. These were included to advance some basic research and are not of concern to most users. For more information, see Chapter III. The user can get back to the ***Model*** worksheet from either the ***Results*** or ***D(k)*** worksheets by depressing the button ***Back to Model*** on each respective worksheet.

APPENDIX B. PROPORTIONAL TRACING TOOLS

If damage aggregation to a target follows a proportional mechanism, then the proportion $D(k)$ of the target damaged by k hits satisfies the two relationships

$$D(k) = D(k-1) + d\{1 - D(k-1)\} = 1 - (1-d)^k,$$

k = 1,2, ... , where $D(0) = 0$, and $D(1) = d$ is the proportion of the target damaged by the first hit.

For an arbitrary damage aggregation mechanism described by a sequence $D(k)$, $k = 0, 1, 2, \ldots$, with $D(0) = 0$, the quantities a_k and d_k, $k = 0, 1, 2, \ldots$, are defined implicitly by their roles in the relationships

$$D(k) = D(k-1) + a_k\{1 - D(k-1)\} = 1 - (1-d_k)^k.$$

The quantity

$$a_k = \frac{D(k) - D(k-1)}{1 - D(k-1)}$$

is the *damage aggregation rate* resulting from the k^{th} hit on the target. It is the proportion of the undamaged target that is damaged by the k^{th} hit. The quantity

$$d_k = 1 - \sqrt[k]{(1 - D(k)}$$

is the *index of proportionality* after the k^{th} hit. It reflects what the proportionality parameter d would be if the damage from k hits had been achieved through a proportional aggregation mechanism. The sequence of proportionality indices $\{d_k,\ k = 1,\ 2,\ \ldots\}$ is the *proportionality trace.*

It follows that

$$1 - (1 - d_{k-1})^{k-1} - a_k(1 - d_{k-1})^{k-1} = D(k) = 1 - (1 - d_k)^k,$$

and that

$$(1 - d_k)^k = (1 - a_k)(1 - d_{k-1})^{k-1},$$

k = 1, 2, ... Applying the preceding equation recursively leads to the factorial like relationship

$$(1 - d_k)^k = (1 - a_k)(1 - a_{k-1})\ldots(1 - a_1),$$

k = 1, 2, ... From here it can be shown (for example by taking logarithms of both sides) that if the damage aggregation rates a_k approach a limit as the number of hits on the targets k approaches infinity, then the proportionality indices d_k approach the same limit.

APPENDIX C. SCENARIOS

This Appendix is the detailed description of the 8 different scenarios analyzed.

Scenario 1

CELL	w_i	r_i	Hardness (c)
1	0.2	0.2	1
2	0.2	0.2	1
3	0.2	0.2	1
4	0.2	0.2	1
5	0.2	0.2	1

Scenario 2

CELL	w_i	r_i	Hardness (c)
1	0.0	0.2	1
2	0.25	0.2	1
3	0.25	0.2	1
4	0.25	0.2	1
5	0.25	0.2	1

Scenario 3

CELL	w_i	r_i	Hardness (c)
1	0.1	0.2	1
2	0.2	0.2	1
3	0.25	0.2	1
4	0.15	0.2	1
5	0.3	0.2	1

Scenario 4

CELL	w_i	r_i	Hardness (c)
1	0.2	0.2	2
2	0.2	0.2	2
3	0.2	0.2	2
4	0.2	0.2	2
5	0.2	0.2	2

Scenario 5

CELL	w_i	r_i	Hardness (c)
1	0.6	0.2	2
2	0.1	0.2	2
3	0.1	0.2	2
4	0.1	0.2	2
5	0.1	0.2	2

Scenario 6

CELL	w_i	r_i	Hardness (c)
1	0.6	0.6	1
2	0.1	0.1	1
3	0.1	0.1	1
4	0.1	0.1	1
5	0.1	0.1	1

Scenario 7

CELL	w_i	r_i	Hardness (c)
1	0.6	0.6	2
2	0.1	0.1	2
3	0.1	0.1	2
4	0.1	0.1	2
5	0.1	0.1	2

Scenario 8

CELL	w_i	r_i	Hardness (c)
1	0.2	0.2	see below
2	0.2	0.2	
3	0.2	0.2	
4	0.2	0.2	
5	0.2	0.2	

Each cell's hardness in Scenario 8 was equal and modeled as follows:

P(Cell i damaged | j =1 hit) = 0.2
P(Cell i damaged | j =2 hits) = 0.4
P(Cell i damaged | j =3 hits) = 0.6
P(Cell i damaged | j =4 hits) = 0.8
P(Cell i damaged | j =5 hits) = 1.0

Scenario 9

CELL	w_i	r_i	Hardness (c)
1	0.2	0.2	see below
2	0.2	0.2	
3	0.2	0.2	
4	0.2	0.2	
5	0.2	0.2	

Each cell's hardness in Scenario 9 was equal and modeled as follows:

P(Cell i damaged | j =1 hit) = 0.0
P(Cell i damaged | j =2 hits) = 0.4
P(Cell i damaged | j =3 hits) = 0.6
P(Cell i damaged | j =4 hits) = 0.8
P(Cell i damaged | j =5 hits) = 1.0

APPENDIX D. TESTING THE TDA

The following are the results of the final 18 runs of the TDA. Runs 10 through 18 were Scenarios 1 through 9 run to determine the Expected Damage. The Salvo Size was 10 and the probability of hit was 0.6. Runs 19 through 27 were Scenarios 1 through 8 run to a desired Expected Damage. Each scenario was run to the $E(D)$ calculated in runs 9 through 16. The number of warheads required to achieve that level of damage was compared for accuracy.

Runs with Salvo Size = 10

Run	Scenario	Salvo Size	$E(D)$
10	1	10	.721499
11	2	10	.721499
12	3	10	.721499
13	4	10	.341725
14	5	10	.341725
15	6	10	.777636
16	7	10	.601207
17	8	10	.239912
18	9	10	.163957

Runs To Desired $E(D)$

Run	Scenario	Desired $E(D)$	Salvo Size
19	1	.7214	10
20	2	.7214	10
21	3	.7214	10
22	4	.3417	10
23	5	.3417	10
24	6	.7776	10
25	7	.6012	10
26	8	.2399	10
27	9	.1639	10

APPENDIX E. RESULTS OF THE 100 WARHEAD RUNS

The results for each of the nine scenarios are enclosed. The salvo size is 100 and the probability the target is hit is 0.6. Once the TDA calculated D(k) as 1, the run was considered complete. Data is provided only for the relevant portion of the data.

Scenario 1

E(D) = .99998

k	D(k)	d_k	a_k
0	0	0	0
1	0.2	0.2	0.2
2	0.36	0.2	0.2
3	0.49	0.2	0.2
4	0.59	0.2	0.2
5	0.67	0.2	0.2
6	0.74	0.2	0.2
7	0.79	0.2	0.2
8	0.83	0.2	0.2
9	0.87	0.2	0.2
10	0.89	0.2	0.2
11	0.91	0.2	0.2
12	0.93	0.2	0.2
13	0.95	0.2	0.2
14	0.96	0.2	0.2
15	0.96	0.2	0.2
16	0.97	0.2	0.2
17	0.98	0.2	0.2
18	0.98	0.2	0.2
19	0.99	0.2	0.2
20	0.99	0.2	0.2
21	0.99	0.2	0.2
22	0.99	0.2	0.2
23	0.99	0.2	0.2
24	1	0.2	0.2
25	1	0.2	0.2
26	1	0.2	0.2
27	1	0.2	0.2
28	1	0.2	0.2
29	1	0.2	0.2
30	1	0.2	0.2
31	1	0.2	0.2
32	1	0.2	0.2
33	1	0.2	0.2
34	1	0.2	0.2
35	1	0.2	0.2
36	1	0.2	0.2

Scenario 1

k	D(k)	d_k	a_k
37	1	0.2	0.2
38	1	0.2	0.2
39	1	0.2	0.2
40	1	0.2	0.2
41	1	0.2	0.2
42	1	0.2	0.2
43	1	0.2	0.2
44	1	0.2	0.2
45	1	0.2	0.2
46	1	0.2	0.2
47	1	0.2	0.2
48	1	0.2	0.2
49	1	0.2	0.2
50	1	0.2	0.2
51	1	0.2	0.2

Scenario 2

$E(D) = .99998$

k	D(k)	d_k	a_k
0	0	0	0
1	0.2	0.2	0.2
2	0.36	0.2	0.2
3	0.49	0.2	0.2
4	0.59	0.2	0.2
5	0.67	0.2	0.2
6	0.74	0.2	0.2
7	0.79	0.2	0.2
8	0.83	0.2	0.2
9	0.87	0.2	0.2
10	0.89	0.2	0.2
11	0.91	0.2	0.2
12	0.93	0.2	0.2
13	0.95	0.2	0.2
14	0.96	0.2	0.2
15	0.96	0.2	0.2
16	0.97	0.2	0.2
17	0.98	0.2	0.2
18	0.98	0.2	0.2
19	0.99	0.2	0.2
20	0.99	0.2	0.2
21	0.99	0.2	0.2
22	0.99	0.2	0.2
23	0.99	0.2	0.2
24	1	0.2	0.2
25	1	0.2	0.2
26	1	0.2	0.2
27	1	0.2	0.2
28	1	0.2	0.2
29	1	0.2	0.2
30	1	0.2	0.2
31	1	0.2	0.2
32	1	0.2	0.2
33	1	0.2	0.2
34	1	0.2	0.2
35	1	0.2	0.2
36	1	0.2	0.2

Scenario 2

k	D(k)	d_k	a_k
37	1	0.2	0.2
38	1	0.2	0.2
39	1	0.2	0.2
40	1	0.2	0.2
41	1	0.2	0.2
42	1	0.2	0.2
43	1	0.2	0.2
44	1	0.2	0.2
45	1	0.2	0.2
46	1	0.2	0.2
47	1	0.2	0.2
48	1	0.2	0.2
49	1	0.2	0.2
50	1	0.2	0.2
51	1	0.2	0.2

Scenario 3

$E(D) = .99998$

k	D(k)	d_k	a_k
0	0	0	0
1	0.2	0.2	0.2
2	0.36	0.2	0.2
3	0.49	0.2	0.2
4	0.59	0.2	0.2
5	0.67	0.2	0.2
6	0.74	0.2	0.2
7	0.79	0.2	0.2
8	0.83	0.2	0.2
9	0.87	0.2	0.2
10	0.89	0.2	0.2
11	0.91	0.2	0.2
12	0.93	0.2	0.2
13	0.95	0.2	0.2
14	0.96	0.2	0.2
15	0.96	0.2	0.2
16	0.97	0.2	0.2
17	0.98	0.2	0.2
18	0.98	0.2	0.2
19	0.99	0.2	0.2
20	0.99	0.2	0.2
21	0.99	0.2	0.2
22	0.99	0.2	0.2
23	0.99	0.2	0.2
24	1	0.2	0.2
25	1	0.2	0.2
26	1	0.2	0.2
27	1	0.2	0.2
28	1	0.2	0.2
29	1	0.2	0.2
30	1	0.2	0.2
31	1	0.2	0.2
32	1	0.2	0.2
33	1	0.2	0.2
34	1	0.2	0.2
35	1	0.2	0.2
36	1	0.2	0.2

Scenario 3

k	D(k)	d_k	a_k
37	1	0.2	0.2
38	1	0.2	0.2
39	1	0.2	0.2
40	1	0.2	0.2
41	1	0.2	0.2
42	1	0.2	0.2
43	1	0.2	0.2
44	1	0.2	0.2
45	1	0.2	0.2
46	1	0.2	0.2
47	1	0.2	0.2
48	1	0.2	0.2
49	1	0.2	0.2
50	1	0.2	0.2
51	1	0.2	0.2

Scenario 4

E(D) = .99996

k	D(k)	d_k	a_k
0	0	0	0
1	0	0	0
2	0.04	0.02	0.04
3	0.1	0.04	0.07
4	0.18	0.05	0.09
5	0.26	0.06	0.1
6	0.34	0.07	0.11
7	0.42	0.08	0.12
8	0.5	0.08	0.13
9	0.56	0.09	0.13
10	0.62	0.09	0.14
11	0.68	0.1	0.14
12	0.73	0.1	0.15
13	0.77	0.11	0.15
14	0.8	0.11	0.15
15	0.83	0.11	0.16
16	0.86	0.12	0.16
17	0.88	0.12	0.16
18	0.9	0.12	0.16
19	0.92	0.12	0.16
20	0.93	0.13	0.17
21	0.94	0.13	0.17
22	0.95	0.13	0.17
23	0.96	0.13	0.17
24	0.97	0.13	0.17
25	0.97	0.13	0.17
26	0.98	0.14	0.17
27	0.98	0.14	0.17
28	0.98	0.14	0.17
29	0.99	0.14	0.17
30	0.99	0.14	0.18
31	0.99	0.14	0.18
32	0.99	0.14	0.18
33	0.99	0.14	0.18
34	1	0.15	0.18
35	1	0.15	0.18
36	1	0.15	0.18

Scenario 4

k	D(k)	d_k	a_k
37	1	0.15	0.18
38	1	0.15	0.18
39	1	0.15	0.18
40	1	0.15	0.18
41	1	0.15	0.18
42	1	0.15	0.18
43	1	0.15	0.18
44	1	0.15	0.18
45	1	0.15	0.18
46	1	0.15	0.18
47	1	0.16	0.18
48	1	0.16	0.18
49	1	0.16	0.19
50	1	0.16	0.18
51	1	0.16	0.19
52	1	0.16	0.19
53	1	0.16	0.19
54	1	0.16	0.19
55	1	0.16	0.19
56	1	0.16	0.19
57	1	0.16	0.18
58	1	0.16	0.2
59	1	0.16	0.19
60	1	0.16	0.2
61	1	0.16	0.19
62	1	0.16	0.19
63	1	0.16	0.22

Scenario 5

$E(D) = .99996$

k	D(k)	d_k	a_k
0	0	0	0
1	0	0	0
2	0.04	0.02	0.04
3	0.1	0.04	0.07
4	0.18	0.05	0.09
5	0.26	0.06	0.1
6	0.34	0.07	0.11
7	0.42	0.08	0.12
8	0.5	0.08	0.13
9	0.56	0.09	0.13
10	0.62	0.09	0.14
11	0.68	0.1	0.14
12	0.73	0.1	0.15
13	0.77	0.11	0.15
14	0.8	0.11	0.15
15	0.83	0.11	0.16
16	0.86	0.12	0.16
17	0.88	0.12	0.16
18	0.9	0.12	0.16
19	0.92	0.12	0.16
20	0.93	0.13	0.17
21	0.94	0.13	0.17
22	0.95	0.13	0.17
23	0.96	0.13	0.17
24	0.97	0.13	0.17
25	0.97	0.13	0.17
26	0.98	0.14	0.17
27	0.98	0.14	0.17
28	0.98	0.14	0.17
29	0.99	0.14	0.17
30	0.99	0.14	0.18
31	0.99	0.14	0.18
32	0.99	0.14	0.18
33	0.99	0.14	0.18
34	1	0.15	0.18
35	1	0.15	0.18
36	1	0.15	0.18

Scenario 5

k	D(k)	d_k	a_k
37	1	0.15	0.18
38	1	0.15	0.18
39	1	0.15	0.18
40	1	0.15	0.18
41	1	0.15	0.18
42	1	0.15	0.18
43	1	0.15	0.18
44	1	0.15	0.18
45	1	0.15	0.18
46	1	0.15	0.18
47	1	0.16	0.18
48	1	0.16	0.18
49	1	0.16	0.19
50	1	0.16	0.19
51	1	0.16	0.19
52	1	0.16	0.19
53	1	0.16	0.19
54	1	0.16	0.19
55	1	0.16	0.19
56	1	0.16	0.19
57	1	0.16	0.19
58	1	0.16	0.19
59	1	0.16	0.19
60	1	0.16	0.2
61	1	0.16	0.2
62	1	0.16	0.19
63	1	0.16	0.21

Scenario 6

E(D) = .999177

k	D(k)	d_k	a_k
0	0	0	0
1	0.4	0.4	0.4
2	0.58	0.35	0.3
3	0.67	0.31	0.21
4	0.72	0.27	0.16
5	0.76	0.25	0.13
6	0.78	0.23	0.11
7	0.81	0.21	0.11
8	0.83	0.2	0.1
9	0.84	0.19	0.1
10	0.86	0.18	0.1
11	0.87	0.17	0.1
12	0.89	0.17	0.1
13	0.9	0.16	0.1
14	0.91	0.16	0.1
15	0.92	0.15	0.1
16	0.93	0.15	0.1
17	0.93	0.15	0.1
18	0.94	0.14	0.1
19	0.95	0.14	0.1
20	0.95	0.14	0.1
21	0.96	0.14	0.1
22	0.96	0.14	0.1
23	0.96	0.14	0.1
24	0.97	0.13	0.1
25	0.97	0.13	0.1
26	0.97	0.13	0.1
27	0.98	0.13	0.1
28	0.98	0.13	0.1
29	0.98	0.13	0.1
30	0.98	0.13	0.1
31	0.98	0.13	0.1
32	0.99	0.13	0.1
33	0.99	0.12	0.1
34	0.99	0.12	0.1
35	0.99	0.12	0.1
36	0.99	0.12	0.1

Scenario 6

k	D(k)	d_k	a_k
37	0.99	0.12	0.1
38	0.99	0.12	0.1
39	0.99	0.12	0.1
40	0.99	0.12	0.1
41	0.99	0.12	0.1
42	1	0.12	0.1
43	1	0.12	0.1
44	1	0.12	0.1
45	1	0.12	0.1
46	1	0.12	0.1
47	1	0.12	0.1
48	1	0.12	0.1
49	1	0.12	0.1
50	1	0.12	0.1
51	1	0.12	0.1
52	1	0.12	0.1
53	1	0.12	0.1
54	1	0.12	0.1
55	1	0.11	0.1
56	1	0.11	0.1
57	1	0.11	0.1
58	1	0.11	0.1
59	1	0.11	0.1
60	1	0.11	0.1
61	1	0.11	0.1
62	1	0.11	0.1
63	1	0.11	0.1
64	1	0.11	0.1
65	1	0.11	0.1
66	1	0.11	0.1
67	1	0.11	0.1
68	1	0.11	0.1
69	1	0.11	0.1
70	1	0.11	0.1
71	1	0.11	0.1
72	1	0.11	0.1
73	1	0.11	0.1
74	1	0.11	0.1
75	1	0.11	0.1

Scenario 6

k	D(k)	d_k	a_k
76	1	0.11	0.1
77	1	0.11	0.1
78	1	0.11	0.1
79	1	0.11	0.1
80	1	0.11	0.1
81	1	0.11	0.1
82	1	0.11	0.1
83	1	0.11	0.1
84	1	0.11	0.1
85	1	0.11	0.1
86	1	0.11	0.09
87	1	0.11	0.1
88	1	0.11	0.1
89	1	0.11	0.09
90	1	0.11	0.1
91	1	0.11	0.1
92	1	0.11	0.1
93	1	0.11	0.1
94	1	0.11	0.1
95	1	0.11	0.1
96	1	0.11	0.1
97	1	0.11	0.09
98	1	0.11	0.09
99	1	0.11	0.09
100	1	0.11	0.09

Scenario 7

E(D) = .993931

k	D(k)	d_k	a_k
0	0	0	0
1	0	0	0
2	0.22	0.12	0.22
3	0.4	0.16	0.23
4	0.51	0.16	0.19
5	0.58	0.16	0.14
6	0.62	0.15	0.1
7	0.65	0.14	0.07
8	0.67	0.13	0.06
9	0.69	0.12	0.05
10	0.7	0.11	0.05
11	0.72	0.11	0.05
12	0.74	0.11	0.06
13	0.75	0.1	0.06
14	0.77	0.1	0.06
15	0.78	0.1	0.06
16	0.79	0.09	0.06
17	0.81	0.09	0.06
18	0.82	0.09	0.07
19	0.83	0.09	0.07
20	0.84	0.09	0.07
21	0.85	0.09	0.07
22	0.86	0.09	0.07
23	0.87	0.09	0.07
24	0.88	0.09	0.07
25	0.89	0.09	0.07
26	0.9	0.08	0.07
27	0.91	0.08	0.07
28	0.91	0.08	0.07
29	0.92	0.08	0.08
30	0.93	0.08	0.08
31	0.93	0.08	0.08
32	0.94	0.08	0.08
33	0.94	0.08	0.08
34	0.95	0.08	0.08
35	0.95	0.08	0.08
36	0.95	0.08	0.08

Scenario 7

k	D(k)	d_k	a_k
37	0.96	0.08	0.08
38	0.96	0.08	0.08
39	0.96	0.08	0.08
40	0.97	0.08	0.08
41	0.97	0.08	0.08
42	0.97	0.08	0.08
43	0.98	0.08	0.08
44	0.98	0.08	0.08
45	0.98	0.08	0.08
46	0.98	0.08	0.08
47	0.98	0.08	0.08
48	0.98	0.08	0.08
49	0.99	0.08	0.08
50	0.99	0.08	0.08
51	0.99	0.08	0.08
52	0.99	0.08	0.09
53	0.99	0.08	0.09
54	0.99	0.08	0.09
55	0.99	0.08	0.09
56	0.99	0.08	0.09
57	0.99	0.08	0.09
58	0.99	0.08	0.09
59	0.99	0.08	0.09
60	0.99	0.08	0.09
61	0.99	0.08	0.09
62	1	0.08	0.09
63	1	0.08	0.09
64	1	0.08	0.09
65	1	0.08	0.09
66	1	0.08	0.09
67	1	0.08	0.09
68	1	0.08	0.09
69	1	0.08	0.09
70	1	0.08	0.09
71	1	0.08	0.09
72	1	0.08	0.09
73	1	0.08	0.09
74	1	0.08	0.09
75	1	0.08	0.09

Scenario 7

k	D(k)	d_k	a_k
76	1	0.08	0.09
77	1	0.08	0.09
78	1	0.08	0.09
79	1	0.08	0.09
80	1	0.08	0.09
81	1	0.08	0.09
82	1	0.08	0.09
83	1	0.08	0.09
84	1	0.08	0.09
85	1	0.08	0.09
86	1	0.08	0.09
87	1	0.08	0.09
88	1	0.08	0.09
89	1	0.08	0.09
90	1	0.09	0.09
91	1	0.09	0.09
92	1	0.09	0.09
93	1	0.09	0.09
94	1	0.09	0.09
95	1	0.09	0.09
96	1	0.09	0.09
97	1	0.09	0.09
98	1	0.09	0.09
99	1	0.09	0.09
100	1	0.09	0.09

Scenario 8

E(D) = .998591

k	D(k)	d_k	a_k
0	0	0	0
1	0.04	0.04	0.04
2	0.08	0.04	0.04
3	0.12	0.04	0.04
4	0.16	0.04	0.05
5	0.2	0.04	0.05
6	0.24	0.04	0.05
7	0.28	0.05	0.05
8	0.32	0.05	0.06
9	0.36	0.05	0.06
10	0.4	0.05	0.06
11	0.44	0.05	0.06
12	0.48	0.05	0.07
13	0.51	0.05	0.07
14	0.55	0.06	0.07
15	0.58	0.06	0.08
16	0.62	0.06	0.08
17	0.65	0.06	0.08
18	0.68	0.06	0.09
19	0.71	0.06	0.09
20	0.73	0.06	0.09
21	0.76	0.07	0.09
22	0.78	0.07	0.1
23	0.8	0.07	0.1
24	0.82	0.07	0.1
25	0.84	0.07	0.1
26	0.86	0.07	0.11
27	0.88	0.07	0.11
28	0.89	0.08	0.11
29	0.9	0.08	0.11
30	0.91	0.08	0.12
31	0.92	0.08	0.12
32	0.93	0.08	0.12
33	0.94	0.08	0.12
34	0.95	0.08	0.12
35	0.95	0.08	0.12
36	0.96	0.09	0.13

Scenario 8

k	D(k)	d_k	a_k
37	0.97	0.09	0.13
38	0.97	0.09	0.13
39	0.97	0.09	0.13
40	0.98	0.09	0.13
41	0.98	0.09	0.13
42	0.98	0.09	0.13
43	0.99	0.09	0.14
44	0.99	0.09	0.14
45	0.99	0.1	0.14
46	0.99	0.1	0.14
47	0.99	0.1	0.14
48	0.99	0.1	0.14
49	0.99	0.1	0.14
50	0.99	0.1	0.14
51	1	0.1	0.14
52	1	0.1	0.15
53	1	0.1	0.15
54	1	0.1	0.15
55	1	0.1	0.15
56	1	0.11	0.15
57	1	0.11	0.15
58	1	0.11	0.15
59	1	0.11	0.15
60	1	0.11	0.15
61	1	0.11	0.15
62	1	0.11	0.15
63	1	0.11	0.15
64	1	0.11	0.15
65	1	0.11	0.16
66	1	0.11	0.16
67	1	0.11	0.16
68	1	0.11	0.16
69	1	0.11	0.16
70	1	0.12	0.16
71	1	0.12	0.16
72	1	0.12	0.16
73	1	0.12	0.16
74	1	0.12	0.16
75	1	0.12	0.16

Scenario 8

k	D(k)	d_k	a_k
76	1	0.12	0.16
77	1	0.12	0.16
78	1	0.12	0.17
79	1	0.12	0.17
80	1	0.12	0.16
81	1	0.12	0.17
82	1	0.12	0.17
83	1	0.12	0.18
84	1	0.12	0.18
85	1	0.12	0.18
86	1	0.13	0.17

Scenario 9

$E(D) = .998583$

k	D(k)	d_k	a_k
0	0	0	0
1	0	0	0
2	0.02	0.01	0.02
3	0.04	0.01	0.03
4	0.08	0.02	0.04
5	0.12	0.02	0.04
6	0.16	0.03	0.05
7	0.21	0.03	0.05
8	0.25	0.04	0.06
9	0.3	0.04	0.06
10	0.34	0.04	0.07
11	0.39	0.04	0.07
12	0.43	0.05	0.07
13	0.48	0.05	0.08
14	0.52	0.05	0.08
15	0.56	0.05	0.08
16	0.59	0.05	0.08
17	0.63	0.06	0.09
18	0.66	0.06	0.09
19	0.69	0.06	0.09
20	0.72	0.06	0.1
21	0.75	0.06	0.1
22	0.77	0.07	0.1
23	0.8	0.07	0.1
24	0.82	0.07	0.1
25	0.84	0.07	0.11
26	0.86	0.07	0.11
27	0.87	0.07	0.11
28	0.89	0.07	0.11
29	0.9	0.08	0.12
30	0.91	0.08	0.12
31	0.92	0.08	0.12
32	0.93	0.08	0.12
33	0.94	0.08	0.12
34	0.95	0.08	0.12
35	0.95	0.08	0.13
36	0.96	0.09	0.13

Scenario 9

k	D(k)	d_k	a_k
37	0.96	0.09	0.13
38	0.97	0.09	0.13
39	0.97	0.09	0.13
40	0.98	0.09	0.13
41	0.98	0.09	0.13
42	0.98	0.09	0.14
43	0.99	0.09	0.14
44	0.99	0.09	0.14
45	0.99	0.1	0.14
46	0.99	0.1	0.14
47	0.99	0.1	0.14
48	0.99	0.1	0.14
49	0.99	0.1	0.14
50	0.99	0.1	0.14
51	1	0.1	0.14
52	1	0.1	0.15
53	1	0.1	0.15
54	1	0.1	0.15
55	1	0.1	0.15
56	1	0.11	0.15
57	1	0.11	0.15
58	1	0.11	0.15
59	1	0.11	0.15
60	1	0.11	0.15
61	1	0.11	0.15
62	1	0.11	0.15
63	1	0.11	0.15
64	1	0.11	0.15
65	1	0.11	0.16
66	1	0.11	0.16
67	1	0.11	0.16
68	1	0.11	0.16
69	1	0.11	0.16
70	1	0.12	0.16
71	1	0.12	0.16
72	1	0.12	0.16
73	1	0.12	0.16
74	1	0.12	0.16
75	1	0.12	0.16

Scenario 9

k	D(k)	d_k	a_k
76	1	0.12	0.16
77	1	0.12	0.16
78	1	0.12	0.17
79	1	0.12	0.17
80	1	0.12	0.17
81	1	0.12	0.17
82	1	0.12	0.17
83	1	0.12	0.17
84	1	0.12	0.18
85	1	0.12	0.18
86	1	0.13	0.17

APPENDIX F. MACRO CODE

The Visual Basic Code used to develop the macros used in the Tactical Decision Aid are enclosed.

```
Public Cellrow As Integer
Public Cellcol As Integer
Public CellNum As Integer

Sub Auto_Open()
' This macro puts the user on the Target sheet and shows the initial dialog  box when the file is opened.
        Sheets("SheetTarget").Select
        DialogSheets("DialogStart").Show
End Sub

Sub Startok_click()
' This macro is designed to execute the desired actions presented to the user  in the initial dialog box.
        If (DialogSheets("DialogStart").OptionButtons("Reinitialize Option").Value = xlOn) Then
                DialogSheets("DialogStart").Hide
                DialogSheets("ReinitializeOption").Show
                If (DialogSheets("ReinitializeOption").OptionButtons("CellOn").Value = xlOn) Then
                        DialogSheets("ReinitHardness").Show
        End If
        ElseIf (DialogSheets("DialogStart").OptionButtons("Continue Option").Value = xlOn) Then
                DialogSheets("DialogStart").Hide
        End If
End Sub

Sub ReInitOk_Click()
' This macro is executed when the user chooses to reinitialize the target.  It will either turn all the cells off
' or on.
Dim ClearIndex As Integer

        If (DialogSheets("ReinitializeOption").OptionButtons("CellOff").Value = xlOn) Then
                Sheets("SheetTarget").Range("A3:J12").Interior.ColorIndex = 3
                Sheets("SheetHardness").Range("B5:CW14").Value = "0"
                Sheets("SheetHardness").Range("B20:CW20").Value = "-99"
                Sheets("SheetValues").Range("A3:J12").Value = "-99"
                Sheets("SheetHardness").Range("B24:CW24").Value = "0"

        ElseIf (DialogSheets("ReinitializeOption").OptionButtons("CellOn").Value = xlOn) Then
                Sheets("SheetTarget").Range("A3:J12").Interior.ColorIndex = 6
                Sheets("SheetHardness").Range("B20:CW20").Value = 0.01
                Sheets("SheetValues").Range("A3:J12").Value = 0.01
                Sheets("SheetHardness").Range("B24:CW24").Value = 0.01
                DialogSheets("ReinitializeOption").Hide
        End If
End Sub

Sub CellOk_Click()
' This macro is executed when the user is changing a cells value.
Dim HitOff As Integer

        If (DialogSheets("CellModel").OptionButtons("CellOff").Value = xlOn) Then
                Sheets("SheetTarget").Cells(Cellrow, Cellcol).Interior.ColorIndex = 3
                Sheets("SheetValues").Cells(Cellrow, Cellcol).Value = "-99"
                Sheets("SheetHardness").Cells(20, CellNum + 1).Value = "-99"
                Sheets("SheetHardness").Cells(24, CellNum + 1).Value = "0"
                For HitOff = 1 To 10
```

```
            Sheets("SheetHardness").Cells(HitOff + 4, CellNum + 1).Value = "0"
        Next HitOff

        ElseIf (DialogSheets("CellModel").OptionButtons("CellOn").Value = xlOn) Then
                Sheets("SheetTarget").Cells(Cellrow, Cellcol).Interior.ColorIndex = 6
                Sheets("SheetValues").Cells(Cellrow, Cellcol).Value =
DialogSheets("CellModel").EditBoxes("Edit Box 9").Text
                Sheets("SheetHardness").Cells(20, CellNum + 1).Value =
DialogSheets("CellModel").EditBoxes("Edit Box 9").Text
                Sheets("SheetHardness").Cells(24, CellNum + 1).Value =
DialogSheets("CellModel").EditBoxes("Edit Box PHit").Text
                DialogSheets("CellModel").Hide
        End If
End Sub

Sub ReinitHardnessOK_Click()
' This macro is utilized when the user is modeling cell hardness in the reinitialize phase.
Dim Hit1 As Integer
Dim HitBox1
Dim CellHit As Integer

    For Hit1 = 1 To 10
        HitBox1 = Val(DialogSheets("ReinitHardness").EditBoxes("Box" & Hit1).Text)
        If HitBox1 > 1 Then HitBox = 1
        For CellHit = 2 To 101
                Sheets("SheetHardness").Cells(Hit + 4, CellHit + 1).Value = HitBox
        Next CellHit
    Next Hit1
End Sub

Sub HardnessOk_Click()
' This macro will place the cell hardness into the TDA.
Dim Hit As Integer
Dim HitBox

    For Hit = 1 To 10
        HitBox = Val(DialogSheets("CellHardness").EditBoxes("Box" & Hit).Text)
        If HitBox > 1 Then HitBox = 1
            Sheets("SheetHardness").Cells(Hit + 4, CellNum + 1).Value = HitBox
    Next Hit
End Sub


Sub ReturnOk_click()
' This macro is utilized when the Return to Model button is depressed.
    Sheets("SheetTarget").Select
End Sub
```

```
Sub Button1_click()
' This macro is used when the button over the cell in the target area is  depressed.  If the cell is to be
' turned  on, it displays the Hardness Dialog.
   Cellrow = 3
   Cellcol = 1
   CellNum = 1
   DialogSheets("CellModel").Show
   If (DialogSheets("CellModel").OptionButtons("CellOn").Value = xlOn) Then
      DialogSheets("CellHardness").Show
   End If
End Sub
```

There are 99 other little macros that are identical to this one and are ommitted here.

```
Public NumberOn As Integer
Public TotalValue
Public TotalProb
Public ExpectedDamage As Single
Public NumReps As Integer

Sub Button101_Click()
' This macro determines how many cells are on.  It also ensures the sum of both
' the relative cell values and the probabilty a cell is hit sum to 1.0.
' If there is an error, a message will appear.  If all is ok, the dialog box
' appears to allow the user to Determine E(D).

Dim CellIndex As Integer

        NumberOn = 0
        TotalValue = 0
        TotalProb = 0
        Sheets("Sheet5").Range("A3:B102").Value = " "
        For CellIndex = 2 To 101
                If Sheets("SheetHardness").Cells(20, CellIndex).Value >= 0 Then
                                NumberOn = NumberOn + 1
                                Sheets("Sheet5").Cells(NumberOn + 2, 1).Value = NumberOn
                                Sheets("Sheet5").Cells(NumberOn + 2, 2).Value = CellIndex - 1
                                TotalValue = TotalValue + Val(Sheets("SheetHardness").Cells(20,
CellIndex).Value)
                                TotalProb = TotalProb + Val(Sheets("SheetHardness").Cells(24,
CellIndex).Value)
                End If
        Next CellIndex
        Sheets("SheetTarget").Cells(17, 10).Value = TotalValue
        Sheets("SheetTarget").Cells(19, 10).Value = TotalProb
        TotalValue = Val(Sheets("SheetTarget").Cells(17, 10).Value)
        TotalProb = Val(Sheets("SheetTarget").Cells(19, 10).Value)
        If (TotalProb = 1 And TotalValue = 1) Then
                Sheets("Sheet4").Range("B4:CX4").Value = " "
                Sheets("Sheet4").Cells(3, 1).Value = "k = 0"
                For CellIndex = 1 To NumberOn
                                Sheets("Sheet4").Cells(3, CellIndex + 1).Value = 0
                Next CellIndex
                DialogSheets("Run").Show
```

```
        End If
        If TotalValue <> 1 Then
                DialogSheets("ErrorValue").EditBoxes("Value").Text = TotalValue
                DialogSheets("ErrorValue").Show
        End If
        If TotalProb <> 1 Then
                DialogSheets("ErrorProb").EditBoxes("Prob").Text = TotalProb
                DialogSheets("ErrorProb").Show
        End If
End Sub


Sub RunOk_Click()
' This macro performs the calculations to Determine E(D).

Dim SalvoSize As Integer
Dim BombIndex As Integer
Dim CellNow As Integer
Dim DCounter As Integer
Dim WorkingCell As Integer
Dim bij As Single
Dim deltak As Single
Dim ri As Single
Dim oneminri As Single
Dim phit As Single
Dim Comb
Dim Comb1
Dim Dk As Single
Dim wi As Single
Dim ED As Single
Dim PNeqK As Single
Dim d As Single
Dim dkmin1 As Single
Dim ak As Single
Dim salvopower As Integer

        SalvoSize = Val(DialogSheets("Run").EditBoxes("Edit Box 5").Text)
        phit = Val(DialogSheets("Run").EditBoxes("Edit Box 7").Text)
         ED = 0
        For BombIndex = 1 To SalvoSize
                Sheets("Sheet4").Cells(BombIndex + 3, 1).Value = BombIndex
                Dk = 0
                For CellNow = 1 To NumberOn
                                If CellNow > 1 Then Sheets("Sheet4").Cells(2, CellNow + 1) = CellNow
                                WorkingCell = Val(Sheets("Sheet5").Cells(CellNow + 2, 2).Value)
                                deltak = 0
                                ri = Val(Sheets("SheetHardness").Cells(24, (WorkingCell + 1)).Value)
                                wi = Val(Sheets("SheetHardness").Cells(20, (WorkingCell + 1)).Value)
                                For DCounter = 1 To BombIndex
                                If DCounter <= 10 Then
                                        bij = Val(Sheets("SheetHardness").Cells((DCounter + 4),
(WorkingCell + 1)).Value)
                                Else
                                        bij = 1
```

```
                                End If
                                Comb = Application.Combin(BombIndex, DCounter)
                                deltak = deltak + (bij * Comb * Application.Power(ri, DCounter) *
 Application.Power((1 - ri), (BombIndex - DCounter)))
                                Next DCounter
                                Sheets("Sheet4").Cells(BombIndex + 3, CellNow + 1).Value = deltak
                                Dk = Dk + deltak * wi
                Next CellNow
                Sheets("D(k)").Cells(BombIndex + 3, 1).Value = BombIndex
                Sheets("D(k)").Cells(BombIndex + 3, 2).Value = Application.Round(Dk, 2)
                 Sheets("D(k)").Cells(BombIndex + 3, 30).Value = Dk
                If Dk < 0.99999 Then
                        d = Application.Round((1 - (Application.Power((1 - Dk), (1 / BombIndex)))), 2)
                Else
                        d = 1
                End If
                Sheets("D(k)").Cells(BombIndex + 3, 3).Value = d
                dkmin1 = Val(Sheets("D(k)").Cells(BombIndex + 2, 30).Value)
                If Dk < 0.99999 Then
                        ak = (Dk - dkmin1) / (1 - dkmin1)
                Else
                        ak = 0
                End If
                Sheets("D(k)").Cells(BombIndex + 3, 4).Value = Application.Round(ak, 2)
                Comb1 = Application.Combin(SalvoSize, BombIndex)
                PNeqK = Comb1 * Application.Power(phit, BombIndex) * Application.Power((1 - phit),
(SalvoSize - BombIndex))
                ED = ED + Dk * PNeqK
        Next BombIndex
        Sheets("Results").Cells(1, 3).Value = ED
        Sheets("Results").Cells(3, 3).Value = SalvoSize
        Sheets("Results").Select
End Sub

Sub Button102_Click()
' This macro determines how many cells are on.  It also ensures the sum of both
' the relative cell values a :d the probabilty a cell is hit sum to 1.0.
' If there is an error, a message will appear.  If all is ok, the dialog box
' appears to allow the user to Run to Desired E(D).

Dim CellIndex As Integer
        NumberOn = 0
        TotalValue = 0
        TotalProb = 0
        Sheets("Sheet5").Range("A3:B102").Value = " "
        For CellIndex = 2 To 101
                If Sheets("SheetHardness").Cells(20, CellIndex).Value >= 0 Then
                                NumberOn = NumberOn + 1
                                Sheets("Sheet5").Cells(NumberOn + 2, 1).Value = NumberOn
                                Sheets("Sheet5").Cells(NumberOn + 2, 2).Value = CellIndex - 1
                                TotalValue = TotalValue + Val(Sheets("SheetHardness").Cells(20,
CellIndex).Value)
                                TotalProb = TotalProb + Val(Sheets("SheetHardness").Cells(24,
CellIndex).Value)
```

```
            End If
        Next CellIndex
        Sheets("SheetTarget").Cells(17, 10).Value = TotalValue
        Sheets("SheetTarget").Cells(19, 10).Value = TotalProb
        TotalValue = Val(Sheets("SheetTarget").Cells(17, 10).Value)
        TotalProb = Val(Sheets("SheetTarget").Cells(19, 10).Value)
        If TotalProb = 1 And TotalValue = 1 Then
            Sheets("Sheet4").Range("B4:CX4").Value = " "
            Sheets("Sheet4").Cells(3, 1).Value = "k = 0"
            For CellIndex = 1 To NumberOn
                    Sheets("Sheet4").Cells(3, CellIndex + 1).Value = 0
            Next CellIndex
            DialogSheets("RunToE(D)").Show
        End If
        If TotalValue <> 1 Then
            DialogSheets("ErrorValue").EditBoxes("Value").Text = TotalValue
            DialogSheets("ErrorValue").Show
        End If
        If TotalProb <> 1 Then
            DialogSheets("ErrorProb").EditBoxes("Prob").Text = TotalProb
            DialogSheets("ErrorProb").Show
        End If
End Sub

Sub RunEDOk_Click()
' This macro performs the Run to Desired E(D) calculations.
'

Dim DesED As Single
Dim SalvoSize As Integer
Dim BombIndex As Integer
Dim CellNow As Integer
Dim DCounter As Integer
Dim WorkingCell As Integer
Dim bij As Single
Dim deltak As Single
Dim ri As Single
Dim phit As Single
Dim Comb
Dim Comb1
Dim Dk As Single
Dim wi As Single
Dim ED As Single
Dim PNeqK As Single
Dim d As Single
Dim dkmin1 As Single
Dim ak As Single

        DesED = Val(DialogSheets("RunToE(D)").EditBoxes("Edit Box 6").Text)
        phit = Val(DialogSheets("RunToE(D)").EditBoxes("Edit Box 7").Text)
        ED = 0
        SalvoSize = 0
        Do While ED < DesED
            ED = 0
            SalvoSize = SalvoSize + 1
```

```
        For BombIndex = 1 To SalvoSize
            Sheets("Sheet4").Cells(BombIndex + 3, 1).Value = BombIndex
            Dk = 0
            For CellNow = 1 To NumberOn
            If CellNow > 1 Then Sheets("Sheet4").Cells(2, CellNow + 1) = CellNow
            WorkingCell = Val(Sheets("Sheet5").Cells(CellNow + 2, 2).Value)
            deltak = 0
            ri = Val(Sheets("SheetHardness").Cells(24, (WorkingCell + 1)).Value)
            wi = Val(Sheets("SheetHardness").Cells(20, (WorkingCell + 1)).Value)
            For DCounter = 1 To BombIndex
                If DCounter <= 10 Then
                        bij = Val(Sheets("SheetHardness").Cells((DCounter +
4), (WorkingCell + 1)).Value)
                Else
                        bij = 1
                End If
                Comb = Application.Combin(BombIndex, DCounter)
                deltak = deltak + (bij * Comb * Application.Power(ri, DCounter) *
Application.Power((1 - ri), (BombIndex - DCounter)))
            Next DCounter
            Sheets("Sheet4").Cells(BombIndex + 3, CellNow + 1).Value = deltak
            Dk = Dk + deltak * wi
            Next CellNow
        Sheets("D(k)").Cells(BombIndex + 3, 1).Value = BombIndex
        Sheets("D(k)").Cells(BombIndex + 3, 2).Value = Application.Round(Dk, 2)
        Sheets("D(k)").Cells(BombIndex + 3, 30).Value = Dk
        If Dk < 1 Then
                d = Application.Round((1 - (Application.Power((1 - Dk), (1 /
BombIndex)))), 2)
        Else
                d = 0
        End If
        Sheets("D(k)").Cells(BombIndex + 3, 3).Value = d
        dkmin1 = Val(Sheets("D(k)").Cells(BombIndex + 2, 30).Value)
        ak = (Dk - dkmin1) / (1 - dkmin1)
        Sheets("D(k)").Cells(BombIndex + 3, 4).Value = Application.Round(ak, 2)
        Comb1 = Application.Combin(SalvoSize, BombIndex)
        PNeqK = Comb1 * Application.Power(phit, BombIndex) * Application.Power((1 -
phit), (SalvoSize - BombIndex))
        ED = ED + Dk * PNeqK
      Next BombIndex
    Loop
    Sheets("Results").Cells(1, 3).Value = ED
    Sheets("Results").Cells(3, 3).Value = SalvoSize
    Sheets("Results").Select
End Sub
```

LIST OF REFERENCES

1. Esary, J.D., *Studies on Damage Aggregation for Weapons Salvos*, Technical Report NPS55-90-16, Naval Postgraduate School, July 1990.

2. Esary, J.D., *Studies on Damage Aggregation for Weapons Salvos II*, Technical Report NPSOR-92-007, Naval Postgraduate School, December 1991.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3.	Professor J.D. Esary (Code OR/EY) Operations Analysis Dept. Naval Postgraduate School Monterey CA 93943	2
4.	Professor L.R. Whitaker (Code OR/WH) Operations Analysis Dept. Naval Postgraduate School Monterey CA 93943	2
5.	LT Jeffrey D. Varady USS Austin (LPD-4) FPO AE 09564-1707	2